AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method for controlling a memory in a digital system,
comprising the steps of:

(a) dividing the memory into a plurality of fixed sized memory blocks, wherein each
memory block has a same size as that of a neighboring memory block;

(b) defining at least one of the memory blocks as a compression/decompression region;

(c) assigning compression priorities to the rest of the memory blocks except the at least
one of the memory blocks defined as the compression/decompression region; and,

(d) making the memory blocks to deal with an external data received according to an
external command, and carrying out compression/decompression of data required in the dealing
with the external data at the compression/decompression region according to the compression
priorities; and

(e) recording a number of access times data in each memory block is accessed,
measuring access frequencies of the memory blocks based on the recorded number of access
times, and resetting the assigned compression priorities.

2. (Currently Amended) A The method as claimed in claim 1, wherein the compression
priorities are set based on the access frequencies of the memory blocks.

3. (Currently Amended) The A method as claimed in claim 1, wherein the dealing with
data in the step (d) includes the steps of data insertion, data erasure, data updating, and data
reading.

4. (Currently Amended) The ~~A~~ method as claimed in claim ~~1~~3, wherein the ~~step of~~ data insertion includes ~~the steps of;~~:

(a) comparing a size of data to be inserted in the memory to ~~an~~ empty memory blocks obtained by compression of the memory blocks,

(b) inserting the data in the empty memory blocks when there are empty memory blocks as large as the data size to be inserted therein as a result of the comparison, and

(c) upon completion of the data insertion, finishing the ~~step of~~ data insertion when a presently ~~remained~~ remaining number of empty memory blocks ~~are~~ is greater than a preset threshold value, and selecting a memory block to be compressed from the ~~remained~~ remaining memory blocks according to the compression priorities when the presently ~~remained~~ remaining number of empty memory blocks ~~are~~ is greater than the preset threshold value.

5. (Currently Amended) The ~~A~~ method as claimed in claim 4, wherein ~~the step of~~ selecting a memory block to be compressed presently in the step (c) starts from a moment starting to use a last empty memory block for the data insertion, or from a moment the preset threshold value is exceeded, with reference to the compression priorities.

6. (Currently Amended) The ~~A~~ method as claimed in claim 1, wherein the data in ~~the~~ a memory block selected for compression is accessible normally during compression of data.

7. (Currently Amended) The ~~A~~ method as claimed in claim 1, further comprising ~~the step of~~ changing references indicating the data in the compressed memory blocks into first starting addresses of the compressed memory blocks, so that ~~the~~ a presently accessing block is identified to be the compressed memory block when the data in the compressed memory block is indirectly accessed through the references.

8. (Currently Amended) The ~~A~~ method as claimed in claim ~~1~~4, further comprising ~~the step of~~ coming into an error processing state if there is no more space in the memory for accommodating the data to be inserted even after all of the memory blocks are compressed as a result of the comparison.

9. (Currently Amended) The ~~A~~ method as claimed in claim ~~1~~3, wherein the ~~step of~~ reading includes ~~a step of~~ data reading a data in a decompressed memory block after decompressing a compressed memory block at the compression/decompression region, if the memory block being accessed presently is a compressed block.

10. (Currently Amended) The ~~A~~ method as claimed in claim ~~1~~4, wherein the ~~step of erasing~~ data erasure includes ~~the steps of;~~:

(a) determining the data to be erased of being a data stored in the compression /decompression region,

5

(b) if it is determined that the data to be erased is a data stored, not in the compression /decompression region, but in the empty memory blocks as a result of the determination, ~~and~~ erasing the data,

(c) if it is determined that the data to be erased is a data stored in the compression /decompression region as a result of the determination, calculating a memory size occupied by the data to be erased in each data block in the compression/decompression region,

(d) comparing an occupied memory size in each memory block in the compression /decompression region and a threshold value of the occupied memory size, and

(e) erasing the compressed data and finishing the ~~erasing step~~ data erasure if the occupied memory size in each memory block calculated for each memory block is smaller than the threshold value of the occupied memory size as a result of the comparison, and decompressing the data if the occupied memory size in each memory block calculated for each memory block is greater than the threshold value of the occupied memory size.

11. (Currently Amended) The ~~A~~ method as claimed in claim 10, further ~~including the step of;~~ comprising:

comparing a number of empty memory blocks of the memory to the preset threshold value of ~~the~~ a respective empty memory block before the compressed memory block is decompressed, and

decompressing the compressed data only when the number of empty memory blocks of the memory is greater than the preset threshold value of the empty memory block.

6

12. (Currently Amended) The A method as claimed in claim 10, wherein the data in the

a memory block is accessible normally until the erasing step is finished completely.

13. (Currently Amended) The A method as claimed in claim 3+, wherein the step of data

updating includes the steps of;

(a) determining the data to be updated of being a data stored in the compression

/decompression region, or in a general memory block,

(b) if it is determined that the data to be updated is a data stored in the memory block as

a result of the determination, updating the data,

(c) if it is determined that the data to be updated is a data stored in the compression

/decompression region as a result of the determination, determining the data to be updated of

being a variable size type,

(d) if it is determined that the data to be updated is not a data of the variable size type as

a result of the determination, decompressing the compressed data temporarily and updating the

data to be updated, and

(e) if it is determined that the data to be updated is a data of the variable size type as a

result of the determination, assigning a new memory block, updating the data to be updated, and

erasing an existing data.

14. (Currently Amended) The A method as claimed in claim 13, wherein the data in the

a memory block selected during the step of data updating is accessible normally during the

updating of data.

15. (Canceled).

7